



Mika Arffman

## **AUTOMAATTINEN OVIKELLO**

# **AUTOMAATTINEN OVIKELLO**

Mika Arffman  
Opinnäytetyö  
Kevät 2013  
Tietotekniikan koulutusohjelma  
Oulun seudun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu  
Tietotekniikan koulutusohjelma, ohjelmistokehityksen sv

---

Tekijä: Mika Arffman

Opinnäytetyön nimi: Automaattinen ovikello

Työn ohjaaja: Pertti Heikkilä

Työn valmistumislukukausi ja -vuosi: Kevät 2013

Sivumäärä: 27

---

Opinnäytetyön aiheena oli toteuttaa Max Technologies Oy:lle automaattinen ovikello, jonka toiminta perustuu RFID-tekniikkaan. Työn tavoitteena oli antaa opinnäytetyön tilaajalle tietoa RFID-tekniikasta ja sen käytöstä osana toimivaa sovellusta. Lopputuloksena oli tarkoitus saada toimiva ovikellojärjestelmä, joka soittaa ovikelloa henkilön saapuessa toimitilaan, käyttää henkilön tunnistukseen RFID-tekniikkaa ja tervehtii henkilöä puhesyntetisaattorin avulla. Jatkossa yritys voisi käyttää opinnäytetyössä saatua tietoa muihin RFID-tekniikkaa hyödyntäviin sovelluksiin esimerkiksi tavaroiden seurantaan ja logistiikkaan liittyvissä sovelluksissa.

Opinnäytetyö aloitettiin perehtymällä RFID-lukijoihin ja niihin liittyvään tekniikkaan. Tarkoituksena oli löytää automaattiseen ovikelloon sopiva RFID-lukija, joka pystyisi lukemaan RFID-tunnisteen useamman metrin päästä. Työssä perehdyttiin myös valoportteihin ja niiden toimintaperiaatteisiin, jotta automaattinen ovikello saisi tiedon siitä milloin henkilö saapuu toimitilaan. Opinnäytetyöhön sisältyi Windows-alustalle toteutettava sovellus, joka ohjaa edellä mainittujen laitteiden toimintaa ja koko automaattisen ovikellon toiminnallisuutta.

Opinnäytetyön tuloksena saatiin toteutettua automaattinen ovikello, jonka toteutus antoi opinnäytetyön tilaajalle tietoa RFID-tekniikasta ja sen hyödyntämisestä.

---

Asiasanat: ohjelmistokehitys, C#, RFID

## **ALKULAUSE**

Tämä opinnäytetyö on tehty Max Technologies Oy:lle kevään 2013 aikana.

Haluan kiittää opinnäytetyön tilaajaa mielenkiintoisesta aiheesta. Haluan kiittää kaikkia yrityksen työntekijöitä, jotka ovat auttaneet opinnäytetyön toteutuksessa. Kiitos myös ohjausryhmään kuuluville henkilöille.

Oulussa 10.5.2013

Mika Arffman

# SISÄLLYS

TIIVISTELMÄ	3
ALKULAUSE	4
SISÄLLYS	5
LYHENTEET	6
1 JOHDANTO	7
2 RFID-TEKNIikka	8
2.1 Tunnisteet ja käyttökohteet	8
2.2 Taajuusalueet	9
2.2.1 LF	9
2.2.2 HF	10
2.2.3 UHF	10
2.2.4 Mikroaallot	10
2.3 RFID-järjestelmän komponentit	11
3 JÄRJESTELMÄN MÄÄRITELMÄ	12
3.1 Henkilön tunnistaminen	12
3.2 Valoportti	12
3.3 Toimintaperiaate	14
4 C#-SOVELLUS	15
4.1 SharpDevelop 4.3	15
4.2 C#	17
4.3 Käyttöliittymä	17
4.4 Sarjaporttisyhteys	18
4.5 RFID-lukijan protokolla	19
4.6 RFID-tunnisteen lukeminen	20
4.7 RFID-tunnisteiden lukemisen lopettaminen	22
4.8 RFID-lukijan lämpötilan lukeminen	22
4.9 Puhesyntetisaattori	23
5 TOTEUTUS	24
6 YHTEENVETO	25
LÄHTEET	26

## LYHENTEET

C#	Microsoftin kehittämä ohjelmointikieli
EPC	Electronic Product Code, sähköinen tuotekoodi
HF	High Frequency, taajuusalue välillä 3–30 MHz
LF	Low Frequency, taajuusalue välillä 30–300 kHz
NFC	Near Field Communication
RFID	Radio Frequency Identification, radiotaajuinen etätunnistus
UHF	Ultra High Frequency, taajuusalue välillä 0,3–3 GHz
USB	Universal Serial Bus

# 1 JOHDANTO

Opinnäytetyö tehtiin vuonna 2007 perustetulle Max Technologies Oy:lle, joka on keskittynyt tuottamaan kattavia ratkaisuja työn- ja kalustohallintaan. Max Technologies Oy:n päätoimipaikka sijaitsee Oulussa, mutta yrityksen toimistoja sijaitsee myös Etelä-Suomessa ja yhteistyökumppaneita Saksassa ja Espanjassa. (1.)

Työn tavoitteena on kehittää yritykselle automaattinen ovikello, joka tunnistaa yrityksen toimitiloihin saapuvan henkilön. Henkilön tunnistamiseen käytetään RFID-korttia, jonka sisältämä tieto on lisätty yrityksen palvelimella sijaitsevaan taustajärjestelmään. Tunnistuksen jälkeen sovellus tervehtii saapuvaa henkilöä puhesyntetisaattorin avulla.

Yrityksellä on aikaisempaa kokemusta NFC-teknologiasta, joten opinnäytetyön tarkoituksena on kartoittaa RFID-teknologian hyötyjä ja mahdollisuuksia yrityksen liiketoiminnan kannalta.

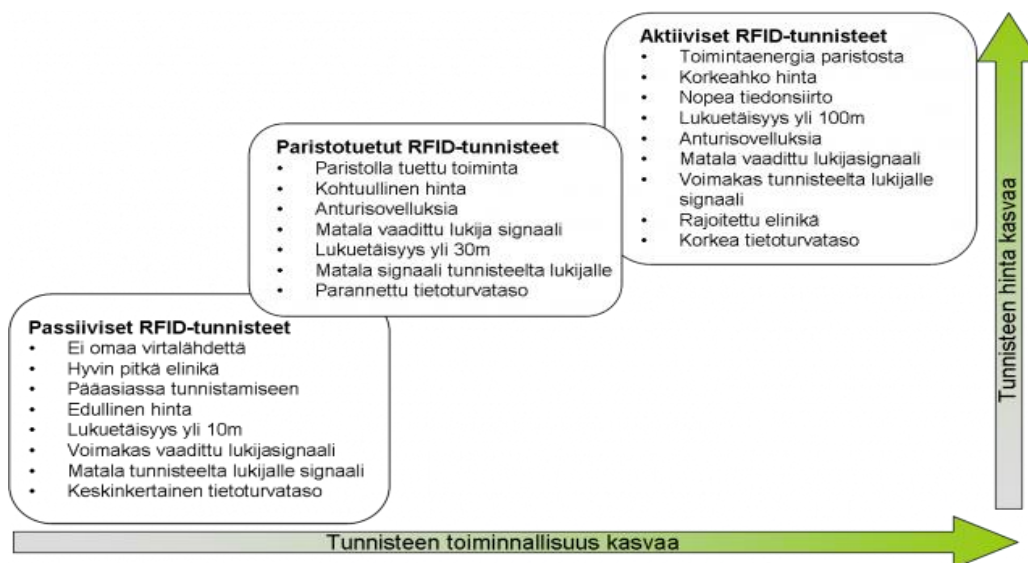
## 2 RFID-TEKNIikka

RFID on yleisnimitys radiotaajuuksilla toimiville tekniikoille, joita käytetään tuotteiden ja asioiden havainnointiin, tunnistamiseen ja yksilöintiin. Teknologia perustuu tiedon tallentamiseen tunnisteeseen ja sen langattomaan lukemiseen radioaaltojen avulla. (2.)

RFID-teknologiaa voidaan verrata viivakoodiin, mutta RFID-tunniste voidaan lukea ilman suoraa katsekontaktia. Lisäksi RFID-tunnisteen sisältöä voidaan muuttaa, kun taas viivakoodi on tulostuksen jälkeen muuttumaton. RFID-tunnisteet kestävät myös paremmin likaisia teollisuusolosuhteita kuin tavanomaiset viivakoodit. (2.)

### 2.1 Tunnisteet ja käyttökohteet

RFID-teknologiassa käytetään kolmea erilaista tunnistetta: passiiviset, puolipassiiviset ja aktiiviset (kuva 1).



KUVA 1. RFID-tunnistetyypit (3)

Passiiviset tunnistet eivät sisällä omaa virtalähdettä. Tunnisteessa on kuparisia silmukoita, jotka muodostavat käämin ja toimivat tunnisteen antennina. Samoin lukijassa on vastaavanlainen silmukka. Lukija ja tunniste keskustelevat moduloimalla oskilloivaa magneettikenttää, joka indusoi



vastaavan vaihtovirran tunnisteeseen käämiin. Tunnisteeseen on oltava tarpeeksi lähellä lukijaa, jotta tunnisteeseen lukeminen onnistuu. (3.)

Paristotuetuissa eli puolipassiivisissa RFID-tunnisteissa on oma virtalähde, jota käytetään vain tietojen lähettämiseen lukijalle. Oma virtalähde puolipassiivisessa tunnisteessa mahdollistaa passiivista tunnistetta pidemmän lukuetaisyyden, mutta muuten puolipassiivinen tunnistee toimii kuten passiivinen tunnistee. (4, s. 10.)

Aktiivisessa RFID-tunnisteessa omaa virtalähdettä voidaan käyttää myös tunnisteeseen laskennan virtalähteenä. Oma virtalähde mahdollistaa tunnisteeseen kirjoittamisen myös silloin, kun tunnistee ei ole lukijan lukuetaisyydellä. (4, s.11.)

RFID-tekniikkaa on hyödynnetty esimerkiksi kulkuavaimissa, matkakorteissa ja eläinten merkitsemisessä. Lisäksi teknologiaa käytetään teollisuudessa osana tuotannon tehostamista ja laadunvalvontaa sekä logistiikassa tavaravirtojen seuraamiseen. (2.)

## **2.2 Taajuusalueet**

RFID-järjestelmiin liittyvät olennaisesti taajuusalueet, joten tunnistee ja lukija on suunniteltu keskustelemaan keskenään radioteitse juuri tietyllä taajuudella. Eri taajuusalueilla fysikaaliset mekanismit ovat erilaisia: LF (Low Frequency)- ja HF (High Frequency) -taajuusalueilla kyseessä on induktiivinen kytkentä, kun taas UHF (Ultra High Frequency)- ja mikroaaltotaajuuksilla radioaallot. Suomessa taajuusalueiden käyttöä kontrolloi Viestintävirasto, joka asettaa rajoitteita ja vaatimuksia myös RFID-laitteistoille. (5.)

### **2.2.1 LF**

LF-taajuusalueella järjestelmät toimivat yleensä 125 kHz:n taajuudella. LF-järjestelmiä ei käytetä paljoa uusissa sovelluskohteissa ja käyttö rajoittuu kulunvalvonnan ja eläintunnistuksen sovelluksiin. (5.)

### **2.2.2 HF**

HF-taajuusalueella standarditaajuus on 13,56 MHz, joka on kansainvälisesti vapaa taajuus. HF-taajuusaluetta käytetään yleensä lähietäisyydellä tunnistamisessa, kuten kulunvalvonnassa. Käytännössä HF-taajuusalueen lukuetaisyydet vaihtelevat viiden senttimetrin ja yhden metrin välillä sovelluksen mukaan. HF-taajuusalueen etuja verrattuna UHF-taajuusalueeseen ovat magneettikentän parempi läpäisykyky vettä sisältäviin aineisiin, esimerkiksi puihin ja ihmisiin, häiriösietoisuus teollisuusympäristöissä, ongelmattomuus heijastusten suhteen, ja helppo lukualueen raja. (5.)

### **2.2.3 UHF**

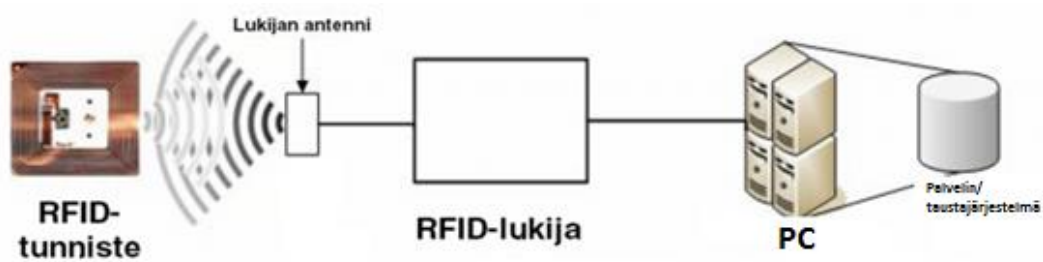
UHF-taajuusalueet vaihtelevat ympäri maailmaa. Esimerkiksi Yhdysvalloissa UHF-taajuusalue on 902–928 MHz, kun taas Euroopassa sallittu taajuusalue on 869 MHz:n ympärillä. UHF-taajuusalueella toimivassa järjestelmässä tunnistajat ja lukijat kommunikoivat radioaaltoja lähettämällä. UHF-taajuusalueella toimivia järjestelmiä käytetään logistiikan sovelluksissa. (5.)

### **2.2.4 Mikroaallot**

Mikroaaltoalueella yleisin taajuus on 2,4 GHz ja mikroaaltoja käytetään enimmäkseen aktiivitunnistuksessa. Mikroaaltoja käytetään esimerkiksi tietullin automaattisen tunnistuksen sovelluksessa. (5.)

### 2.3 RFID-järjestelmän komponentit

RFID-järjestelmään kuuluu RFID-tunniste, RFID-lukija ja taustajärjestelmä (kuva 2). Lukijan antenni lähettää energian tunnisteelle ja vastaanottaa tunnisteelta tulevan informaation ja siirtää sen lukijalle. Lukija lukee informaation ja lähettää sen taustajärjestelmään, joka käsittelee informaation sovelluksen mukaan. Yleensä tunnisteeseen kirjoitetaan vain yksilöivä EPC-sarjanumero ja varsinainen tieto haetaan taustajärjestelmän tietokannasta. (3.)



KUVA 2. RFID-järjestelmä (3)

### 3 JÄRJESTELMÄN MÄÄRITELMÄ

Automaattinen ovikello ilmoittaa ovikellon äänellä yrityksen toimitilaan saapuvasta henkilöstä ja tervehtii tunnistettua henkilöä puhesyntetisaattorin avulla. Järjestelmä muodostuu RFID-lukijasta, RFID-kortista, valoportista ja sovelluksesta.

#### 3.1 Henkilön tunnistaminen

Henkilön tunnistamiseen käytetään RFID-lukijaa, joka lukee saapuvalla henkilöllä olevan RFID-kortin (kuva 4). Tämän jälkeen sovellus tarkistaa, onko kortin sisältämä tunnistus lisätty yrityksen palvelimella sijaitsevaan taustajärjestelmään. Jos tunnistus löytyy taustajärjestelmästä ja henkilö kulkee valoportin läpi, tervehtii automaattinen ovikello saapuvaa henkilöä palvelimen palauttaman tiedon mukaan.

Jos saapuvalla henkilöllä ei ole RFID-tunnistetta tai sitä ei ole lisätty yrityksen palvelimella sijaitsevaan taustajärjestelmään, automaattinen ovikello soittaa pelkän ovikellon äänen henkilön kulkiessa valoportin läpi.



*KUVA 3. RFID-lukija, RFID-lukijan antenni ja RFID-kortit (6)*

#### 3.2 Valoportti

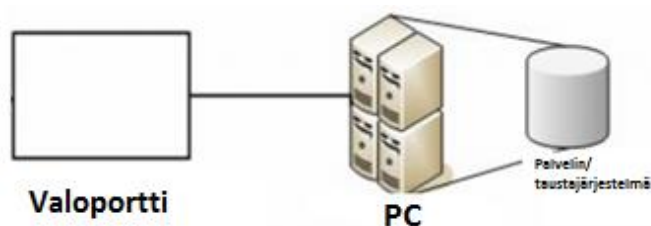
Automaattisessa ovikellossa käytetään infrapuna-LED-valoporttia tunnistamaan henkilöiden kulkua yrityksen toimitilaan (kuva 3). Valoportti muodostuu lähettimestä ja vastaanottimesta, jotka muodostavat väliinsä kolme infrapunavalokeilaa. Valoportti on kytketty tietokoneen USB-porttiin käyttäen

sarjaporttikaapelia ja USB-adapteria. Sarjaporttikaapelissa on käytössä vain kaksi johtoa: sisäänmeno sekä ulostulo. Sarjaporttikaapelissa kulkeva ulostulojohto antaa jännitettä valoportin lähettimessä olevalle releelle ja sisäänmenojohto antaa signaalin sovellukselle releen kytkeytyessä. Signaalin avulla sovellus osaa soittaa ovikelloa oikealla ajanhetkellä ja lisäksi tervehtiä saapuvaa henkilöä, jos RFID-lukija lukee henkilöllä olevan taustajärjestelmään lisätyn RFID-kortin. Lähettimessä oleva rele kytkeytyy, kun saapuva henkilö kulkee valoportin läpi ja katkaisee yhtäaikaaisesti lähettimen ja vastaanottimen välillä olevat kolme infrapunavalokeilaa.



*KUVA 4. Infrapuna-LED-valoportti*

Automaattisessa ovikellossa valoportti on yhdistetty tietokoneeseen käyttäen sarjaporttikaapelia ja USB-adapteria (kuva 5).

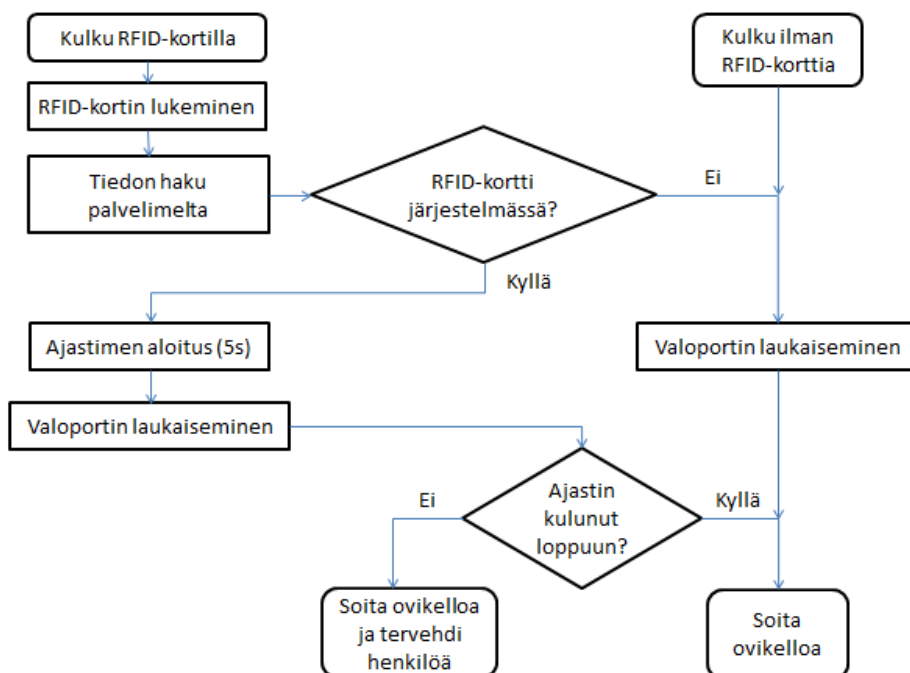


*KUVA 5. Valoportti järjestelmässä (3)*

### 3.3 Toimintaperiaate

RFID-lukija lukee saapuvan henkilön RFID-kortin ja hakee korttiin liitetyt tiedot yrityksen palvelimella sijaitsevasta taustajärjestelmästä. Tämän jälkeen sovellus käynnistää viiden sekunnin mittaisen ajastimen. Saapuvalla henkilöllä on viisi sekuntia aikaa kulkea valoportin läpi, jotta automaattinen ovikello osaa tervehtiä saapuvaa henkilöä puhesyntetisaattorin avulla. Jos henkilöllä kestää yli viisi sekuntia kulkea valoportin läpi, automaattinen ovikello soittaa pelkän ovikellon äänen. Laskurilla pyritään poistamaan väärän henkilön tervehtiminen, koska sovellus poistaa henkilöön liittyvät tiedot muistista ajastimen kuluessa loppuun. Automaattinen ovikello tervehtii saapuvaa henkilöä palvelimen palauttaman tiedon ja puhesyntetisaattorin avulla. (Kuva 6.)

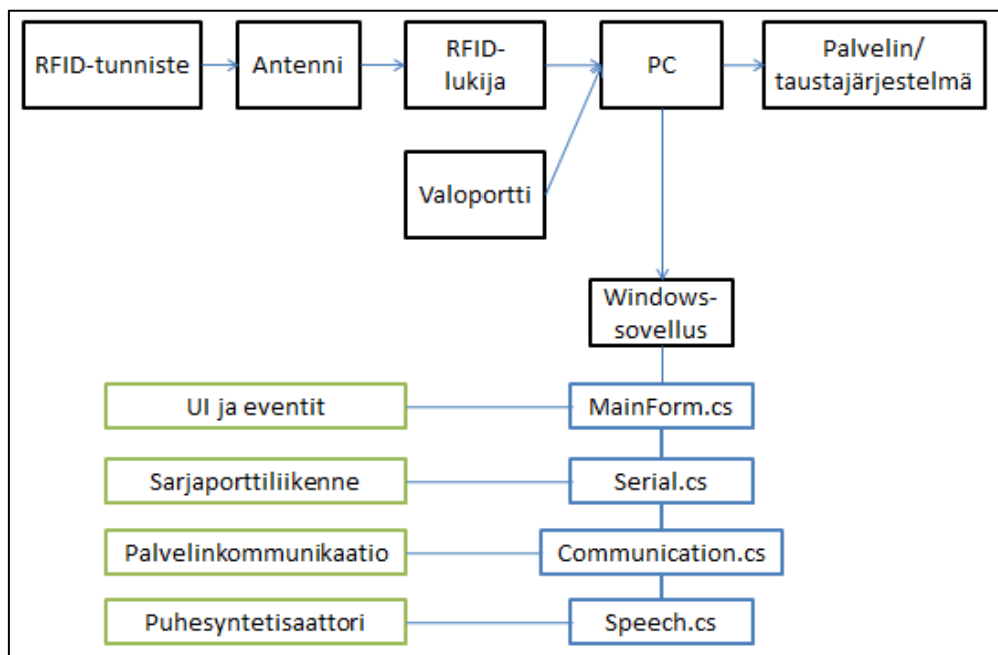
Automaattinen ovikello soittaa vain pelkän ovikellon äänen, jos saapuvalla henkilöllä ei ole RFID-korttia tai sitä ei ole lisätty yrityksen palvelimella sijaitsevaan taustajärjestelmään.



KUVA 6. Automaattisen ovikellon vuokaavio

## 4 C#-SOVELLUS

Automaattiseen ovikelloon kuuluva, Windows-ympäristössä ajettava sovellus toteutettiin SharpDevelop 4.3 -ohjelmalla ja ohjelmointikielenä käytettiin C#:a (C Sharp). Sovellus muodostuu pääosin neljästä luokasta (kuva 7). MainForm-luokka hoitaa käyttöliittymän ja siihen liittyvien komponenttien eventit. Serial-luokka hoitaa sarjaporttiliikenteen, jota käytetään RFID-lukijan ja valoportin kommunikaatioon. Communication-luokka hakee palvelimelta RFID-tunnistetta vastaavat tiedot, joita Speech-luokka käyttää puhesyntetisaattorissa.



KUVA 7. Windows-sovellus automaattisessa ovikellossa

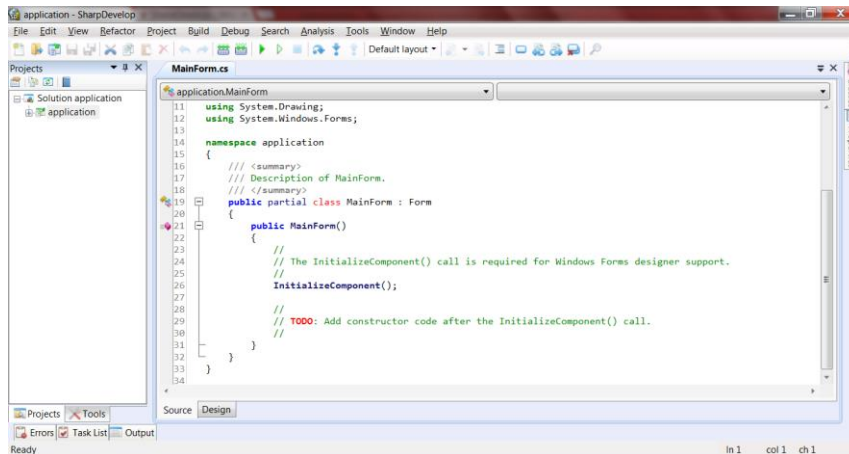
### 4.1 SharpDevelop 4.3

SharpDevelop on ilmainen avoimeen lähdekoodiin perustuva ohjelmointiympäristö Microsoftin .NET:lle ja se tukee ohjelmistokehitystä seuraavilla ohjelmointikielillä: C#, Visual Basic .NET, Boo, F#, IronPython ja IronRuby. (7.)

SharpDevelop suunniteltiin ilmaiseksi ja kevyeksi vaihtoehdoksi Microsoft Visual Studiolle ja se sisältää melkein kaikki Visual Studio Express -version ominaisuudet, kuten projektin hallinnan, koodinmuokkauksen, sovelluksen

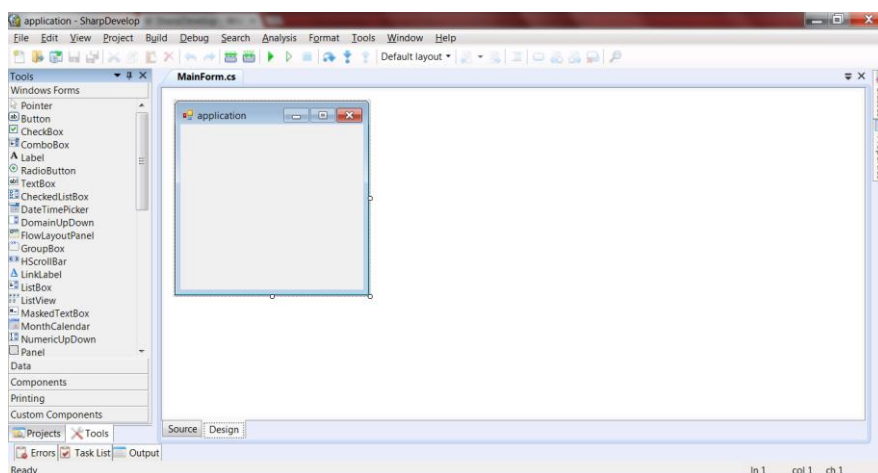
kääntämisen ja virheenetsinnän. Visual Studioon projektit ja kooditiedostot toimivat suoraan SharpDevelopissa, joten se mahdollistaa helpon siirtymisen toiseen ohjelmointiympäristöön. (7.)

SharpDevelopin käyttöliittymä on hyvin samannäköinen kuin Microsoft Visual Studioon käyttöliittymä (kuva 8).



*KUVA 8. SharpDevelopin käyttöliittymä*

SharpDevelop sisältää samoja ominaisuuksia kuin Microsoft Visual Studio, mm. GUI-suunnittelutyökalun, jonka avulla saadaan toteutettua sovelluksen käyttöliittymä helposti ja nopeasti (kuva 9).



*KUVA 9. SharpDevelopin GUI-suunnittelutyökalu*

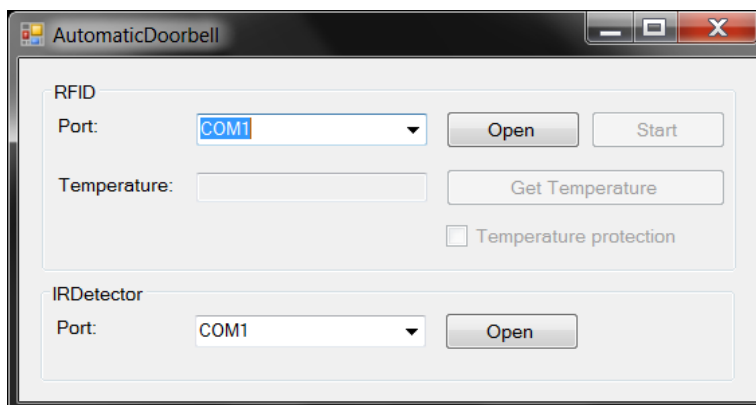


## 4.2 C#

C# on Microsoftin kehittämä ohjelmointikieli .NET-konseptia varten ja se julkaistiin vuoden 2000 kesäkuussa. C# kehitettiin yhdistämään C++:n tehokkuus ja Java-kielen helppokäyttöisyys. C# sai ISO-standardin vuonna 2003. (8.)

## 4.3 Käyttöliittymä

Automaattisen ovikellon käyttö ei sinänsä perustu sovelluksen käyttöliittymään, mutta sen avulla automaattinen ovikello saadaan toimintavalmiuteen. Toteutettu käyttöliittymä on yksinkertainen ja helppokäyttöinen, ja siitä voidaan seurata myös RFID-lukijan lämpötilaa. (Kuva 10.)



KUVA 10. Automaattinen ovikello –sovelluksen käyttöliittymä

Käyttöliittymässä on RFID-lukijalle ja valoportille oma osio, jossa on Open-näppäin sarjaporttiyhteyden avaamiselle. RFID-lukijan osiossa on myös Start-näppäin, josta RFID-lukija aloittaa RFID-tunnisteiden lukemisen. RFID-lukijan lämpötila saadaan haettua Get Temperature -näppäimellä ja lämpötila näytetään omassa tekstikentässä.

Käyttöliittymän RFID-lukijan osiossa voidaan laittaa päälle Temperature protection -toiminto, joka hidastaa automaattisesti RFID-lukijan lukutaaajuutta, kun lukijan lämpötila kohoaa yli 60 °C:een. Kun RFID-lukijan lämpötila kohoaa

yli 85 °C:een, RFID-lukija lopettaa tunnistusten lukemisen ja ilmoittaa poikkeavasta lämpötilasta. (9.)

#### 4.4 Sarjaporttiyhteys

Automaattisessa ovikellossa RFID-lukija ja valoportti ovat liitettyinä tietokoneen USB-porttiin, joten molemmat laitteet kommunikoivat sovelluksen kanssa sarjaporttiyhteyden avulla. RFID-lukijan sarjaporttiyhteyden avaamiseen käytetään muuttujaa, joka luodaan SerialPort-luokasta. Luodulle muuttujalle annetaan sen portin nimi, johon RFID-lukija on liitetty. Muuttujalle asetetaan RFID-lukijan käyttämä baudinopeus, tässä tapauksessa 115 200 bittiä sekunnissa. Ennen portin avaamista muuttujalle asetetaan DataReceived-niminen eventti, jonka spRFID\_DataReceived-metodi käsittelee. Edellä mainittua metodia kutsutaan, kun RFID-lukija lähettää tietoa sarjaporttiin. RFID-lukijan lähettämä tieto käsitellään ja parsitaan spRFID\_DataReceived-metodin aloittamassa säikeessä. Lopuksi sarjaporttiyhteys avataan kutsumalla spRFID-muuttujalla Open-metodia. (Koodiesimerkki 1.)

```
//Asettaa portin nimen
spRFID.PortName = portName;
//Asettaa portin baudinopeuden
spRFID.BaudRate = 115200;
//Asettaa DataReceived-eventin
spRFID.DataReceived += new SerialDataReceivedEventHandler(spRFID_DataReceived);
//Avaa sarjaporttiyhteyden
spRFID.Open();
```

##### *KOODIESIMERKKI 1. Sarjaporttiyhteyden avaaminen*

Sarjaporttiyhteys valoportille luodaan samalla periaatteella kuin RFID-lukijalle, mutta toimiakseen valoportti tarvitsee DTR-signaalin (Data Terminal Ready). Sarjaporttikaapelin DTR-pinni on kytkettynä valoportissa olevaan releeseen, joka saa tarvittavan ohjausjännitteen DTR-signaalin ollessa aktivoituna. (Koodiesimerkki 2.)

```
//Ottaa käyttöön DTR-signaalin (Data Terminal Ready)
spIRDetector.DtrEnable = true;
```

## *KOODIESIMERKKI 2. DTR-signaalin käyttöönotto*

Valoportin sarjaporttimuuttujalle asetetaan PinChanged-eventti, joka kutsuu spIRDetector\_PinChanged-metodia. Valoportin releen kytkeytyessä sarjaportin sisäänmenopinnan tila vaihtuu, jolloin sovellus kutsuu automaattisesti spIRDetector\_PinChanged-metodia. Edellä mainittu metodi aloittaa uuden säikeen, joka tutkii, onko yrityksen palvelimelta saatu tietoja saapuvasta henkilöstä. Sovellus tervehtii saapuvaa henkilöä palvelimen palauttaman tiedon perusteella.

### **4.5 RFID-lukijan protokolla**

Protokolla perustuu UART:iin (Universal Asynchronous Receiver Transmitter) ja USB:iin (Universal Serial Bus) ja se mahdollistaa yhteyden RFID-lukijan ja tietokoneen välillä. (9.)

UART-sarjaliikennepiiri on tyypillisesti mikropiiri, joka muuntaa rinnakkaismuotoista tietoa sarjamuotoiseksi ja päinvastoin. UART-piiriä käytetään usein asynkronisessa sarjamuotoisessa tietoliikenteessä tietokoneen ja oheislaitteen välillä. (10.)

RFID-lukijan ja tietokoneen välillä kulkeva tieto on heksadesimaalimuodossa ja se koostuu alku- ja loppuosasta, pituudesta, komennosta, tiedosta ja tarkistuskoodista (kuva 11).

Header	Length	Command	Data	Check code	End
2 bytes	2 bytes	1 byte	N bytes	1 byte	2 bytes

*KUVA 11. RFID-lukijan tiedonsiirtoon käytettävä rakenne. (9, s.5.)*

Datapaketti alkaa kahdella tavulla, jotka ovat muuttumattomia tavuja: 0xA5 ja 0x5A. Datapaketti loppuu kahteen tavuun, jotka ovat myös muuttumattomia: 0x0d ja 0xda. Datapaketissa oleva pituus on koko tietorakenteen pituus ja se lasketaan seuraavanlaisella kaavalla:  $\text{Length} = 2 + 2 + 1 + N + 1 + 2$ , missä  $N$  on tiedon pituus. Datapaketin komento muodostuu yhdestä tavusta ja jokaiselle toiminnolle on oma komento. Esimerkiksi komennolla 0x82 aloitetaan RFID-tunnisteiden lukeminen. Datapaketin datatavuja käytetään parametreinä ja niiden avulla RFID-lukija voi esimerkiksi palauttaa tiedon siitä, onnistuiko lähetetty komento. Yhden tavun mittainen tarkistuskoodi kertoo datapaketin sisältämän tiedon tavuarvojen eron. (9.)

#### 4.6 RFID-tunnisteen lukeminen

Koska saapuva henkilö voi kulkea RFID-lukijan antennin ohitse millon vain, täytyy lukijan yrittää lukea tunnisteita koko ajan. RFID-lukijalle lähetetty komento aloittaa RFID-lukijassa loputtoman RFID-tunnisteiden lukemisen, joka voidaan lopettaa vain omalla siihen tarkoitetulla komennolla. (Koodiesimerkki 3.)

```
byte[] bytesToSend = new byte[10] {0xA5, 0x5A, 0x00, 0x0A, 0x82, 0x00, 0x00, 0x88, 0x0D, 0x0A};  
spRFID.Write(bytesToSend, 0, bytesToSend.Length);
```

#### *KOODIESIMERKKI 3. RFID-tunnisteen lukeminen*

Lähetettävä data sijoitetaan kymmenpaikkaiseen tavutyypin taulukkoon ja sen jälkeen taulukko kirjoitetaan sarjaporttiin kutsumalla Write-metodia. Write-metodin ensimmäinen parametri on puskuri eli lähetettävä tavutyypin taulukko ja toinen parametri on puskurin offset, joka kertoo, mistä puskurin kohdasta tavujen kirjoittaminen sarjaporttiin aloitetaan. Viimeinen eli kolmas parametri kertoo sarjaporttiin kirjoitettavien tavujen määrän. (Koodiesimerkki 3.)

RFID-lukija kirjoittaa datapaketin sarjaporttiin, kun se on lukenut RFID-tunnisteen. Datapaketti sisältää RFID-tunnisteen EPC-numeron, jonka avulla saapuva henkilö tunnistetaan. (Kuva 12.)

Header		Length		Command	Data		
0xA5	0x5A	0x00	0xxx	0x83	PC (MSB)	PC (LSB)	EPC (MSB)
Data							
EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC
Data					Check code	End	
EPC	EPC	EPC (LSB)	RSSI (MSB)	RSSI (LSB)	Ant Num.	0xxx	0x0D 0x0A

KUVA 12. RFID-lukijan palauttama RFID-tunnisteen tieto (9, s. 47–48.)

Sovellus kutsuu automaattisesti DataReceived-eventtiä, kun RFID-lukija kirjoittaa datapaketin sarjaporttiin. DataReceived-eventin aloittama uusi säie lukee sarjaportista saapuvan datapaketin ja sijoittaa sen tavutyypin taulukkoon. (Koodiesimerkki 4.)

```
byte[] buffer = new byte[spRFID.BytesToRead];
spRFID.Read(buffer, 0, buffer.Length);
```

#### KOODIESIMERKKI 4. Tiedon lukeminen sarjaportista

Datapaketin sisältämä RFID-tunnisteen EPC-numero parsitaan muusta datapaketin sisältämästä tiedosta (koodiesimerkki 5).

```
public string ParseTagID(byte[] data)
{
    byte[] tagID = new byte[TAGID_LENGTH];
    int index = 0;

    for (int i = 7; i < 19; i++)
    {
        tagID[index] = data[i];
        index++;
    }

    return BitConverter.ToString(tagID).Replace("-", "");
}
```

#### KOODIESIMERKKI 5. EPC-numeron parsiminen

ParseTagID-metodissa luodaan 12-paikkainen tavutyyppinen taulukko, johon EPC-numero sijoitetaan käyttäen apuna for-silmukkaa. Metodi palauttaa merkkijono-tyyppisen EPC-numeron ilman väliviivoja. (Koodiesimerkki 5.)

EPC-numeroa vastaavat henkilön tiedot haetaan yrityksen palvelimella sijaitsevasta taustajärjestelmästä käyttäen yrityksen protokollaa.

#### 4.7 RFID-tunnisteiden lukemisen lopettaminen

Kun RFID-tunnisteiden lukeminen halutaan lopettaa, lähetetään RFID-lukijalle lukemisen lopettava komento. RFID-tunnisteiden lukemisen lopettamisen komento lähetetään samalla periaatteella kuin koodiesimerkissä 3 selostettu RFID-tunnisteiden lukeminen. (Koodiesimerkki 6.)

```
byte[] bytesToSend = new byte[8] {0xA5, 0x5A, 0x00, 0x08, 0x8C, 0x84, 0x0D, 0x0A};  
spRFID.Write(bytesToSend, 0, bytesToSend.Length);
```

*KOODIESIMERKKI 6. RFID-tunnisteiden lukemisen lopettaminen*

#### 4.8 RFID-lukijan lämpötilan lukeminen

Jotta RFID-lukijan lämpötila ei nousisi liian korkeaksi, sitä voidaan seurata sovelluksen käyttöliittymästä painamalla Get Temperature -näppäintä. Kun edellä mainittua näppäintä painetaan, sovellus lähettää komennon RFID-lukijalle, joka palauttaa lukijan lämpötilan heksadesimaalimuodossa. Jotta lämpötila saadaan oikeaan muotoon, täytyy lämpötila muuttaa heksadesimaalimuodosto desimaalimuotoon ja tämän jälkeen jakaa luku vielä sadalla. Lämpötila näytetään celsiusasteina sovelluksen käyttöliittymässä sille varatussa tekstikentässä.

Sovelluksen käyttöliittymäosiossa selostettu Temperature protection -ominaisuus voidaan kytkeä päälle sovelluksen käyttöliittymästä. Kun ominaisuus kytketään päälle käyttöliittymästä, sovellus lähettää RFID-lukijalle komennon, joka palauttaa sovellukselle tiedon onnistuiko kyseisen ominaisuuden kytkeminen päälle. Ominaisuuden kytkeminen pois päältä tapahtuu täysin samalla periaatteella, mutta pois kytkemiseen käytetään luonnollisesti omaa komentoa.

## 4.9 Puhesyntetisaattori

Sovelluksen puhesyntetisaattori käyttää äänenä englanninkielistä Microsoft Annaa, joka on oletusäänenä Windows Vistassa ja Windows 7:ssä. (11.)

SpeechSynthesizer-luokan avulla voidaan esimerkiksi puhua tekstiä, numeroita ja päivämääriä, ja myös tekstin puhumisnopeutta voidaan muuttaa. Edellä mainitun luokan avulla sovellus tervehtii saapuvaa henkilöä taustajärjestelmän palauttaman tiedon perusteella. (Koodiesimerkki 7.)

```
speechSynth = new SpeechSynthesizer();
speechSynth.SetOutputToDefaultAudioDevice();

PromptBuilder builder = new PromptBuilder();
builder.AppendText(speech);
speechSynth.SpeakAsync(builder);
```

### *KOODIESIMERKKI 7. Puhuminen puhesyntetisaattorilla*

Aluksi luodaan SpeechSynthesizer-luokasta muuttuja, joka määrittää ulostulon lähetettäväksi oletuksena olevalla äänilaitteelle. Seuraavaksi luodaan PromptBuilder-luokasta muuttuja, johon liitetään puhuttava teksti AppendText-metodin avulla. Esimerkissä puhuttava teksti on sijoitettu speech-nimiseen muuttujaan, joka on merkkijonotyyppinen muuttuja. Lopuksi teksti puhutaan SpeechSynthesizer-luokasta luodun muuttujan ja sen SpeakAsync-metodin avulla. (Koodiesimerkki 7.)

## 5 TOTEUTUS

Järjestelmän toteutus aloitettiin aivan puhtaalta pöydältä, joten mitään ei ollut valmiina. Aluksi oli perehdyttävä RFID-tekniikkaan ja siihen, millainen RFID-lukija sopisi kehitettävään järjestelmään. Koska RFID-tunnisteen lukuetaisyyden täytyi olla useamman metrin, valittiin UHF-taajuudella toimiva RFID-lukija. Järjestelmässä täytyi olla myös jonkinlainen valoportti saapuvan henkilön tunnistukseen, joten päädyttiin infrapunatekniikkaa hyödyntävään valoporttiin.

Tilattuja laitteita odotellessa oli aikaa perehtyä kehitysympäristöihin, joten päädyttiin kehittämään sovellus C Sharp -ohjelmointikielellä Windows-alustalle. Valintaan päädyttiin, koska kehitysympäristöstä löytyi tuki puhesyntetisaattorille ja ohjelmointikieli oli helposti ymmärrettävää.

Valoportin saapuessa täytyi selvittää sen toimintaperiaate ja valoportin yhdistäminen tietokoneeseen. Valoportille tehtiin tarvittavat johdotukset, joten se saatiin toimimaan melko nopeasti. Jonkin verran ongelmia aiheutti kytkentöjen tekeminen tietokoneen ja valoportin välillä, mutta nekin saatiin ajan kanssa toimimaan.

Seuraavaksi oli vuorossa RFID-lukijaan perehtyminen ja sen lisääminen toteutettavaan järjestelmään. Laitteen mukana toimitettiin demo-ohjelma, jolla pystyttiin testaamaan laitteen toimintaa ja ominaisuuksia. RFID-lukijan liittäminen osaksi järjestelmää ei tuottanut kovin paljon vaikeuksia, koska laitteen mukana toimitettiin kattavat dokumentaatiot.

Ohjelmarakenteesta pyrittiin tekemään yksinkertainen, jotta sitä olisi helppo ymmärtää ja sitä voisi hyödyntää jatkokehityksessä ja muissa vastaavanlaisissa sovelluksissa.



## 6 YHTEENVETO

Työn tavoitteena oli kehittää automaattinen ovikello, joka käyttää henkilön tunnistukseen RFID-tekniikkaa. Suurena osa-alueena työssä oli RFID-tekniikkaan tutustuminen ja sen hyödyntäminen automaattisessa ovikellossa. Tavoitteena oli myös toteuttaa Windows-alustalla toimiva sovellus, joka ohjaa automaattisen ovikellon toimintaa. Lisäksi työssä käytettiin valoporttia ja perehdyttiin sen toimintaperiaatteeseen, jotta automaattisesta ovikellosta saatiin täysin toimiva kokonaisuus.

Automaattinen ovikello tervehtii saapuvaa henkilöä puhesyntetisaattorin avulla, jos henkilöllä on saapuessaan taustajärjestelmään lisätty RFID-kortti mukana ja RFID-lukija lukee sen. Vaikka henkilöllä ei olisikaan RFID-korttia, automaattinen ovikello soittaa ovikellon äänen henkilön kulkiessa valoportin läpi.

Ainoa automaattisen ovikellon heikkous on RFID-lukijan käyttämän UHF-taajuusalueen rajoitteet nesteiden ja metallien läheisyydessä. Työssä käytetyllä RFID-lukijalla pystyy lukemaan tunnisteen jopa seitsemän metrin päästä antennista, mutta ihmisen ollessa antennin ja tunnisteen välillä tunnisteen lukeminen vaikeutuu. Tästä syystä automaattisessa ovikellossa käytettyjen RFID-korttien piti olla hyvin antennin luettavissa, esimerkiksi vaatteiden etutaskussa.

Työssä toteutettava automaattinen ovikello saatiin toteutettua aikataulun mukaisesti ja toteutettu järjestelmä tarjoaa hyvät jatkokehitysmahdollisuudet muille alustoille ja järjestelmille. Työssä saatuja RFID-tekniikkaan liittyviä tietoja voidaan hyödyntää esimerkiksi logistiikkaan tai tavaroiden seuraantaan liittyvissä sovelluksissa.

## LÄHTEET

1. Max Technologies Oy. Saatavissa: <https://www.maxtech.fi/index.php/fi/yritys>. Hakupäivä 12.3.2013.
2. Mitä on RFID. Saatavissa: <http://www.rfidlab.fi/rfid-tietoutta>. Hakupäivä 26.3.2013.
3. RFID-tekniikan fysikaaliset perusteet. Saatavissa: <http://www.rfidlab.fi/rfid-tekniikan-perusteet>. Hakupäivä 26.3.2013.
4. Rinta-Runsala, Esa – Tallgren, Markus. 2004. RFID-tekniikan hyödyntäminen asiakkuudenhallinnassa. Saatavissa: <http://www.vtt.fi/inf/julkaisut/muut/2004/rfid-raportti.pdf>. Hakupäivä: 26.3.2013.
5. RFID-tekniikan käyttämät taajuualueet. Saatavissa: <http://www.rfidlab.fi/rfid-tekniikan-k%C3%A4ytt%C3%A4m%C3%A4t-taajuualueet>. Hakupäivä 26.3.2013.
6. Blanco RFID kortti. Saatavissa: <http://www.idcontrol.fi/tuotteet?ItemCategory=8&Item=25>. Hakupäivä 10.4.2013.
7. SharpDevelop. 2013. Saatavissa: <http://en.wikipedia.org/wiki/SharpDevelop>. Hakupäivä 25.4.2013.
8. C sharp. 2013. Saatavissa: [http://fi.wikipedia.org/wiki/C\\_sharp](http://fi.wikipedia.org/wiki/C_sharp). Hakupäivä 16.4.2013.
9. Indy Series UHF RFID Reader Module Communication Protocol. RFID-lukijan PDF-dokumentti.
10. UART. 2013. Saatavissa: <http://fi.wikipedia.org/wiki/UART>. Hakupäivä 24.4.2013.

11. Microsoft text-to-speech voices. 2013. Saatavissa: [http://en.wikipedia.org/wiki/Microsoft\\_text-to-speech\\_voices](http://en.wikipedia.org/wiki/Microsoft_text-to-speech_voices). Hakupäivä: 26.4.2013.
12. SpeechSynthesizer Class. Saatavissa: <http://msdn.microsoft.com/en-us/library/system.speech.synthesis.speechsynthesizer.aspx>. Hakupäivä: 26.4.2013.